

2. (Original) The system of claim 1, wherein the table entries further include an indication of whether an invalid state is activated such that the invalid state for a table entry becomes activated when that table entry contains no useable mapping information.

5

3. (Original) The system of claim 2, wherein the agent does not allow the host to complete the I/O operations with one of the entries if the invalid state for that entry is activated.

10 4. (Original) The system of claim 1, wherein the table entries further include an indication of whether a no-write state is activated such that the no-write state for one of the entries becomes activated when data cannot be written to the storage location contained in that entry.

15 5. (Original) The system of claim 4, wherein the agent does not allow the host to write data to the storage location in one of the entries if the no-write state for that entry is activated.

6. (Original) The system of claim 1, further comprising a communication
20 channel to couple the agent and the controller.

7. (Original) The system of claim 6, wherein the communication channel employs a data transfer protocol to transport messages on the communication channel.
- 5 8. (Original) The system of claim 1, wherein the entries include an offset.
9. (Original) The system of claim 8, wherein the offset includes logic unit number identifier.
- 10 10. (Original) The system of claim 8, wherein the offset includes a block identifier.
11. (Original) The system of claim 10, wherein the entries further includes a segment of virtual disk positions.

12. (Currently Amended) A system for mapping a virtual disk segment to a storage location within a storage device, ~~such that a host issues a I/O operation to an agent and the agent determines said storage location for input/output operations, said system~~ comprising:

5 a host computer to issue an I/O operation identifying a logical unit and a block identifier within the logical unit;

an agent connected to a host computer, the agent having a volatile memory module for storing a first copy of a table having entries to map a virtual disk position to a storage location on the storage devices and a plurality of

10 variables indicating states of the respective entries; and

a controller coupled to the agent, the controller having non-volatile memory for storing a second copy of the table, the controller intermittently causing contents of the first copy of the table to be replaced by contents of the second copy of the table.

15 ~~a table having an entry corresponding to said storage location;~~

~~a plurality of variables indicating states of the entry;~~

~~an offset for the entry, wherein the offset includes a logic unit number identifier and a block identifier; and~~

~~a memory to store the table.~~

20

13. (Original) The system of claim 12, wherein the memory is volatile.

14. (Original) The system of claim 12, wherein said storage location comprises a block of data within the storage device.

15. (Previously Presented) The system of claim 14, wherein the block of data
5 is 1 MB.

16. (Original) The system of claim 12, wherein the agent is coupled to the host.

10 17. (Original) The system of claim 12, wherein the plurality of variables comprise Boolean variable.

18. (Original) The system of claim 12, wherein the states include an invalid state.

15

19. (Original) The system of claim 18, wherein the plurality of variables includes a variable for the invalid state.

20. (Original) The system of claim 12, wherein the states include a no-write
20 state.

21. (Original) The system of claim 20, wherein the plurality of variables includes a variable for the no-write state.

22. (Original) The system of claim 12, wherein the states include a zero state.

5 23. (Original) The system of claim 12, wherein the states include an error state.

Caven & Aghevli LLC
8/4/2005

7

200302185-1

24. (Original) A method for performing an operation on a virtual disk coupled to a host within a network, comprising:

specifying a block on the virtual disk within the operation;

accessing a table mapping the block to a storage location on a storage

5 device;

issuing a corresponding operation to the storage device, wherein the

corresponding operation correlates to the operation on the virtual disk;

completing the corresponding operation; and

presenting the completed corresponding operation to the virtual disk.

10

25. (Original) The method of claim 24, wherein the issuing step includes issuing the corresponding operation from an agent coupled to the host.

26. (Original) The method of claim 24, further comprising updating the table

15 with a persistently-stored table residing in a non-volatile memory.

27. (Original) The method of claim 24, further comprising determining states of the table.

20 28. (Original) The method of claim 24, further comprising sending a fault message when the table is unable to be accessed.

29. (Original) The method of claim 24, further comprising storing the table in a volatile memory.

30. (Previously Presented) The method of claim 24, further comprising
5 receiving updates for the table from a controller.

31-33. Canceled

34. (Original) A computer program product comprising a computer useable medium having computer readable code embodied therein for performing an operation on a virtual disk coupled to a host within a network, the computer program product adapted when run on a computer to effect steps including:
- 5 specifying a block on the virtual disk within the operation;
- accessing a table mapping the block to a storage location on a storage device;
- issuing a corresponding operation to the storage device, wherein the corresponding operation correlates to the operation on the virtual disk;
- 10 completing the corresponding operation; and
- presenting the completed corresponding operation to the virtual disk.

35. Canceled.

36. (Previously Presented) The system of claim 12, wherein said storage location comprises a variable sized group of contiguous virtual disk blocks.

5

37. (Previously Presented) The system of claim 12, wherein said table entry represents a variable sized group of contiguous virtual disk blocks.

38. (Previously Presented) The system of claim 37, wherein said contiguous
10 virtual disk blocks map to contiguous blocks on a storage container.

39. (Previously Presented) The system of claim 12, wherein said table entry comprises a beginning block.

15 40. (Previously Presented) The system of claim 12, wherein said table entry comprises an ending block.

20